Amendments to the Specification:

Please replace the paragraph beginning on page 7, line 6 with the following amended paragraph:

As mentioned above the logic circuits 46, 48 are enabled by an EDSIN signal applied to their Si inputs. The EDSIN signal is generated by a flip-flop 80 formed by two NOR gates 84, 86, the output of which is coupled through an inverter 90. The flip-flop 80 is set to enable the logic circuit 46, 48 by applying a high data strobe write enable DSWE signal to the NOR gate 84. The flip-flop 80 is reset ~~the~~to disable the logic circuits 46, 48 and reset their outputs high either applying an active low BRSTi signal to an inverter 94 or by applying an inactive high ENDSi signal to the NOR gate 86. However, as mentioned above, the ENDSi signal is normally active low during the operation of the data strobe circuit 10, so the NOR gate 86 is normally enabled. A low transitioning BRSTi pulse, which resets the flip-flop 80, is generated at the output of the pulse generator 68 whenever the DSBi signal transitions high. As previously explained, this occurs when the fourth bit of data is latched into the flip-flop 28. However, since the DSWE is normally high when the data strobe circuit 10 is active, these BRSTi pulses do not reset the flip-flop 80 to disable the logic circuits 46, 48. However, when the data strobe circuit 10 is to be disabled for a write operation, the DSWE signal transitions low to allow the BRSTi pulse to be generated when the fourth bit of data has been strobed into the flip-flop 28.

Please replace the paragraph beginning on page 7, line 23 with the following amended paragraph:

The operation of the entire data strobe circuit 10 will now be explained with reference to the timing diagram shown in Figure 3, which shows various signals present in the circuit of Figure 2 over a 150 ns time period as indicated at the top of Figure 3. Figure 3A shows a clock signal that provides the basic timing for a memory device (not shown) containing the data strobe circuit 10 of Figure 1. Figure 3B shows a data strobe signal DS having several pulse pairs each of which is used for strobing 4 bits of data into the memory device. As further shown

in Figure 3B, a pair of noise pulses occur on the DS line starting at about 115 ns. As explained above, the logic circuit 46 is initially enabled so that each DS pulse shown in Figure 3B causes a DSA pulse to be generated, as shown in Figure 3C. This DSA pulse latches the first and second data bits into the flip-flops 22, 24, respectively. When each DSA pulse is generated, the Write2i signal shown in Figure 3J is active low so that the falling edge of the DSA pulse (the rising edge of the DSAi pulse) causes a DSC pulse to be generated at the output of the NOR gate 70, as shown in Figure 3H. Each of these DSC pulses toggles the flip-flop 60, thereby disabling the logic circuit 46 and enabling the logic circuit 48. As a result, the subsequent DS pulse causes a DSB pulse to be generated, as shown In Figure 3D. Each DSB pulse latches the ~~second~~third and ~~third~~fourth data bits into the flip-flops 26, 28, respectively, and causes a DSR pulse to be generated at the output of the NAND-gate 64, as shown in Figure 3I. This DSC pulse resets the flip-flop 60, thereby enabling the logic circuit 46 and disabling the logic circuit 48 so that the subsequent DS pulse generates a DSA pulse rather than a DSB pulse, as explained above.

Please replace the paragraph beginning on page 10, line 17 with the following amended paragraph:

The above-described operation of the SDRAM 100 is controlled by a command decoder ~~168~~104 responsive to command signals received on a control bus 170. These high level command signals, which are typically generated by a memory controller (not shown in Figure 4), are a clock enable signal CKE*, a clock signal CLK, a chip select signal CS*, a write enable signal WE*, a row address strobe signal RAS*, and a column address strobe signal CAS*, which the "*" designating the signal as active low. Various combinations of these signals are registered as respective commands, such as a read command or a write command. The command decoder ~~168~~104 generates a sequence of control signals responsive to the command signals to carry out the function (*e.g.*, a read or a write) designated by each of the command signals. These command signals, and the manner in which they accomplish their respective functions, are conventional. Therefore, in the interest of brevity, a further explanation of these control signals will be omitted.

Please replace the paragraph beginning on page 11, line 1 with the following amended paragraph:

Figure 5 shows a computer system 200 containing the SDRAM 100 of Figure 4. The computer system 200 includes a processor 202 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 202 includes a processor bus 204 that normally includes an address bus, a control bus, and a data bus, which includes the data strobe signal. In addition, the computer system 200 includes one or more input devices 214, such as a keyboard or a mouse, coupled to the processor 202 to allow an operator to interface with the computer system 200. Typically, the computer system 200 also includes one or more output devices 216 coupled to the processor 202, such output devices typically being a printer or a video terminal. One or more data storage devices 218 are also typically coupled to the processor 202 to allow the processor 202 to store data in or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 218 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs). The processor 202 is also typically coupled to cache memory 226, which is usually static random access memory ("SRAM"), and to the SDRAM 100 through a memory controller 230. The memory controller 230 normally includes ~~a~~the control bus ~~236~~106 (Figure 4) and ~~an~~the address bus ~~238~~114 that are coupled to the SDRAM 100. ~~A~~The data bus ~~240~~158 is coupled from the SDRAM 100 to the processor bus 204 either directly (as shown), through the memory controller 230, or by some other means.